VIRTUAL MACHINE LOGBOOK

DIPLOMA PROJECT – SUMMER-FALL 2008

TASKS TO REALIZE

August 13, 2008

STUDENT:	JULIEN POFFET
SUPERVISORS:	FRÉDÉRIC BAPST
	PAOLO CALAFIURA
	OTTAR JOHNSEN
	YUSHU YAO
EXPERT:	PETER KROPF

CONTENTS

INTRODUCTION

(COMMON)

VIRTUAL MACHINES

In computer science, a machine is composed of two main parts: the hardware and the software. The operating system (OS) directly communicates with the hardware, to offer to the applications an interface for access the physical resources.

A virtual machine (VM) is an abstraction of a whole computer: the software and also the hardware part. This means that a virtual machine contains applications, a guest operating system and a virtualized hardware. The guest OS and the applications execute like in a real machine. The difference is that instead of accessing to the physical hardware, the guest OS accesses to the virtualized one. It is the task of the virtual machine monitor (VMM, or hypervisor) to link the virtualized hardware to the real one[1].

There are several advantages of using virtual machines. The most important in this project are:

- a VM contains the entire environment of a machine: the operating system with the environment variables, the applications with their configuration and the user data;
- a VM which is in a given host machine can be serialized, moved and then executed in a different host machine.

A Just enough Operating System (JeOS) is a virtual machine which contains only the minimal components needed to run some specific applications in order to lighten as most as possible the size of the VMs.

ATLAS EXPERIMENT

The Large Hadron Collider (LHC) at the CERN is a particle accelerator which will be used to run physics experiments. ATLAS is the name of one of the detectors in the LHC. During the experiments, it will generate hundreds GB of data which will be distributed over the LHC grid. The LHC grid is a network of computers centers all around the globe that will process and analyze the LHC data. ATLAS is also the name of the software used to run these analyses. It is composed by about 5M lines of codes organized in a thousand packages. This source code is used by the physicists to write ATLAS data processing programs.

ATLAS software is in continuous development: at present hundreds of physicists run analyses on simulation data to tune up the algorithms, fix bugs and add missing features. Sharing the work among physicists and developers can be hard, because of the different working environments. Compiling and running the same program on two machines with different configurations may not produce the same behavior. Because of that, sometimes it is difficult to reproduce on different machines a crash or an analysis which produces an interesting result.

[1] The link may be direct or indirect, i.e. handled by the host operating system.

CERNVM[2]

The goal of the CernVM project is to easily provide downloadable Just enough Operating Systems for the physicists working on the LHC experiments. The virtual machine available on the CernVM website is very light and can be personalized to the needs of the physicist. For example, it is possible to automatically install the components needed to run the ATLAS software.

The goal of using CernVM JeOS is to take the advantages offered by the VMs, to allow the physicists to work with an official up-to-date release of the ATLAS software, and to reduce the software maintenance cost. CernVM is a project which is still in development at the CERN.

PROJECT DESCRIPTION

(COMMON)

CORE FEATURES

We have to develop the Virtual Machine Logbook, an application whose goal is to simplify the share of the work environments used in the ATLAS experiment. The key idea is to profit by the advantages offered by the virtual machines: if the physicists work into virtual machines instead of real machines, it is possible to easily share the work environments by sharing the virtual machines. In this project we suppose that the physicists work into virtual machines.

In this project we have to understand the needs of the physicists who work for the ATLAS experiment and develop a first working version of the Virtual Machine Logbook. This application will then be submitted to a team of testers that will provide a feedback on usability and real-time performance.

The main feature that Virtual Machine Logbook has to provide is the access to the logbook. The logbook is a repository which contains the environments used by the physicists. The environment is a virtual machine. There are three main features to access the Virtual Machine Logbook: add, check out and delete an entry (i.e., a virtual machine).

- The user who is working with a VM on a project can add it into the logbook by a simple command. The VM is then stored into the repository. During his work, the user can do other backups of his machine. All these backups are stored in the logbook. The logbook will then contain a history of the state of the virtual machine used by the physicist.
- If the user needs to restore an older state, he can retrieve the corresponding virtual machine from the logbook. After that, he will be able to work in the same environment which he was using when he added the VM into the repository.
- All the backup history of a virtual machine can be deleted from the repository. The user can also choose to delete just a version of the virtual machine, without deleting all the history.

[2] http://cernvm.cern.ch/cernvm

The virtual machines to store in the logbook will have a big size. For this reason, the Virtual Machine Logbook will probably not always add and store whole virtual machines. We have to find a solution to lighten the size of the entries in the logbook. The Virtual Machine Logbook must also provide an easy way to download project-specific software from the CernVM repositories directly to a virtual machine.

The repository of the Virtual Machine Logbook can be placed in the local machine or in another remote machine. The physicist who needs to do backups of his work will probably store the VMs on his own machine. Using a remote logbook accessible by multiple users may be useful to share the virtual machines. A centralized Virtual Machine Logbook could be used as a central repository which contains the official ATLAS software releases.

Using the Virtual Machine Logbook in a client-server model allowing multiple users to add, check out and delete virtual machine may raise security issues. A virtual machine may contain grid certificates[3], mounted network drives, logins or passwords belonging to the user who added the machine. This kind of information has to be hidden/deleted if somebody checks out the virtual machine elsewhere. To avoid this, Virtual Machine Logbook has to provide to the user a way to choose if his virtual machine is private or public. A private virtual machine will keep the security settings of the user but will not be published for the other users. A public virtual machine will be downloadable by everyone but all the user security settings, network directory or other grid certificate will be removed from the machine.

When a physicist is working inside a guest virtual machine, he probably needs to access some files stored in his host machine. We have to provide a simple way to share file system between physical hosts and guest virtual machines. This file sharing has to be provided for all the virtualization technologies used by CernVM. It must also be easily enabled or disabled.

FURTHER WORK

If time allows, we will add some extensions to the Virtual Machine Logbook. There are some general ideas which could be studied and developed after the core featured described in the last section.

At this moment, the physicists work in their physical machines and not into virtual machines. Before starting work into a virtual machine, they need to move their environment data from the physical host machine to the guest VM. A useful feature would be the creation of a virtual machine from a physical machine. This VM should contain the same software environment of the physical machine, to allow the physicist to continue his work into the virtual machine, without any further configuration.

A virtual machine could have different network configurations. NAT (Network Address Translation) is a possible configuration which may complicate the access to mass storage systems. In particular, it is not possible to open a connection from the outside network. The virtual machines used by the physicists in the ATLAS experiment should be able to access all mass storage systems. It may be necessary to develop tools to allow that.

Data on the grid should be accessible inside a virtual machine. We should investigate the technique to allow that.

---

[3] The grid certificate is required to get an access to the LHC grid.

At present, to retrieve data from the CernVM software repository, the user has to explicitly execute a command. It would be interesting to automatically synchronize the local disk cache with the content of the CernVM repository. This may be realized using a non-static file system.

Another usefully feature that Virtual Machine Logbook may provide is the deployment and management of the virtual machines on a remote server/cluster/cloud. The virtual machines stored in the logbook have to be easily deployed on one or more machines and then started. Then the physicist has to be able to run his analysis on these virtual machines. Once the job is terminated, the physicist has to use a Virtual Machine Logbook command to stop the virtual machines and release the resources which were used by them.

PREFERRED TOOLS AND BOUNDARY CONDITIONS

The Virtual Machine Logbook has to be compatible with the CernVM just enough operating systems. The virtual machines downloadable from the CernVM project web site are built using rBuilder. This tool allows you to prepare virtual machines of all major VM platforms (QEMU/KVM, Parallels, Xen and VMware). The virtualization solutions should work on Linux hosts, but also on Windows and Mac OS X.

The applications for the ATLAS experiment have to be developed using open-source or at least freely available software products, because they are used by a large number of people of different universities and organizations.

WORK TO DO

STUDY

ATLAS started an investigation on Virtual machine technologies, and it is currently collaborating with the CernVM project. The scientists who use the CernVM virtual appliance may work on different project (i.e, ATLAS, ALICE). The CernVM project provides a way to update the generic virtual appliance to a more specific one according to the project on which the scientist is working on. I will study how the CernVM project makes this update in order to know if I can use a similar process to save or restore a virtual machine state.

The physicist develops, tests and executes his analysis by working in an environment. The environment is composed by the operating system, the applications used by the physicist, the environment variables and the rest of the work area. We have to study how to make a "smart" snapshot of this working environment. Making a full snapshot each time will be a bad solution in term of disk space and speed. Our "smart" snapshot will probably consist of saving some part files or copying some part of the environment into files. My part of work about this "smart" snapshot will focus on how I can use a versioning system tool to save and restore these different files. In order to do that, I will make an investigation about the different versioning system tools.

To summarize, these are the principal tasks I have to do for the first part:

- Study the virtualization tools.
  - Study the possibility offered by their API in terms of reading or writing files from/to the virtual machine.

- Study the CernVM project.
    - Read documentation about CernVM and understand how CernVM already supports ATLAS software.
    - Understand how the CernVM virtual machine management interface works and how is done the file transfer from the CernVM repository to the VMs.
    - Study the file system used by the CernVM repository.
- Study the tools which could be useful to get and restore (some parts of) the environment, like file synchronization tools.
- Study the versioning systems which could be adapted for Virtual Machine Logbook.

After this part we will know the requirement of the different ATLAS software and we will have a general idea on the features offered by the different tools which will be used by the Virtual Machine Logbook.

DESIGN

After studying the CernVM project and the versioning tools, I will have enough information to do the design of the Virtual Machine Logbook application.

Firstly, I will specify all the basic functionality that will be offered by the Virtual Machine Logbook. In other word, I will think about how the physicist will add, checkout and delete his virtual machine to/from the Virtual Machine Logbook. The program must also offer the capability to delete a whole entry or a specific version of an entry. Note that the user has different places to launch a Virtual Machine Logbook command. He can do it directly into his virtual machine or he can also do it from his host machine. So I have to specify for each command if it can be executed only from the host machine, the virtual machine or from both.

Secondly, I will specify all the modification that I will perform on the CernVM just enough operating system (JeOS) in order to build a virtual machine that contains all the software needed to allow the user to access the Virtual Machine Logbook from the VM.

The Virtual Machine Logbook will store the virtual machine states as an entry. Basically an entry is a full description of a working environment. The first entry added will be used as a reference by the next ones. The other entry will describe only the difference between the given environment and the reference entry. For instance, a physicist who works on a specific analysis may add every day an entry on the logbook to keep a trace of his work. The first entry is the reference and the next ones will just contain the modifications done from this reference. This is the idea that we refer when we talk about "smart" snapshots. Instead of taking a picture of the whole environment at each time, we take one reference picture and then we pick only the difference between the reference one and the next ones. I have to find a solution to organize efficiently these entries.

The detailed tasks I have to do for the design part are the following:

- Specify all the basic functionalities of the Virtual Machine Logbook.
    - Describe the use cases of the application.
    - Draw the physical model of the application, i.e. the architecture in which the application will run.

    - Define which features will be available inside and outside the virtual machine.
    - Separate the features into components/modules and specify the interfaces between them. This will become the logical model.
    - Map the logical model to the physical model (i.e., where the modules will execute).
- Specify the modification we will apply to the CernVM JeOS.
    - Define how the user will be able to access the Virtual Machine Logbook within the virtual machine.
- Specify the accesses and the transfer from/to the Virtual Machine Logbook.
    - Specify how to manage the entries storage in a centralized/shared repository.
    - Describe how the entries are transferred from/to the Virtual Machine Logbook in both local and remote repository.
    - Describe how to manage the user access rights to a centralized/shared repository.
    - Specify how to connect to the CernVM repository and download project-specific software from it.
    - Design a command line interface to manage the Virtual Machine Logbook
- During this phase of design, I have to choose a versioning system.

Before to start with the realization phase, we have also to know what programming language(s) we will use. It may be necessarily to familiarize with the programming language(s) before to start the development of the Virtual Machine Logbook.

IMPLEMENTATION

After the design, the Virtual Machine Logbook application will be separated into different components/modules. Each module will need and/or provide a service to other modules. So for this part, I have to implement these modules.
Here is a first sketch of all the components/modules I will implement during this phase:

- Realize the Virtual Machine Logbook repository management. The modules to develop have to implement the following tasks:
    - Store the entry in the repository (locally and remotely).
    - Package all the files before transferring them to the Virtual Machine Logbook in order to decrease the network overload.
    - Transfer the files from/to the Virtual Machine Logbook.
    - Allow the user to choose if his machine will be public (sharable) or private
    - Remove a whole entry or just a version of the Virtual Machine Logbook.
    - Check out an entry from the Virtual Machine Logbook to a local or remote machine.
- Realize the additional requirements of the application.
    - Management of the Virtual Machine Logbook from a command line interface.
    - Management of the Virtual Machine Logbook user access.
    - Selection and download of the project-specific software from the CernVM repository.

- Prepare a test virtual machine.
- Realize and perform the test scenarios.
- Finalize the application.
    - Prepare a deployment package and an installation manual of the application.
    - Write a user manual for the application.

TESTS

(COMMON)

We have to prepare some test scenarios to verify if our application correctly runs. These tests will help us during the development to localize bugs and fix them. Each feature offered by the Virtual Machine Logbook has to be tested.

Some test scenarios will be conceived to verify if a single module/component of the application works as we expected, for instance if the entries are correctly structured and filled during the preparation of the entries. Other test scenarios will be realized to check the global behavior of the application, for example the integrity of the environment has to be guarantee after adding and checking out a VM.

ORGANIZATION OF THE PROJECT

(COMMON)

The project is done at the Lawrence Berkeley National Laboratory (LBNL) for a period of 15 weeks. We will work together on the same project, but it is separated into two personal projects and we will be individually evaluated.

KEY DATES

- July 27 – Beginning of the project
- November 7 – End of the project
- November 12 – Back to Switzerland
- November 14-15 – Public presentation at the EIA-FR

WEB SITE

All the documents concerning the project will be posted on our web site:

http://phyweb.lbl.gov/atlaswiki/index.php/VMLogbook

login: guest-vml
password: vmlogbook

The web site contains two separate sections, one for each personal project.

CONTACTS

(COMMON)

STUDENTS

- Andrea Cavalli cavalli86@gmail.com
- Julien Poffet julienpoffet@gmail.com

SUPERVISORS

- Frédéric Bapst EIA-FR – frederic.bapst@hefr.ch
- Ottar Johnsen EIA-FR – ottar.johnsen@hefr.ch
- Paolo Calafiura LBNL – pcalafiura@lbl.gov
- Yushu Yao LBNL – yyao@lbl.gov

EXPERT

- Peter Kropf UniNE – peter.kropf@unine.ch